

National Aeronautics and Space Administration

volume 9, number 3 | Summer 2011



GSFC NEWS



Cover Art by Trina Cox

tech transfer

Automated Software Development Issue

in this issue:

- 2 | From the Chief
- 3 | Tech Transfer Opportunities
- 4 | Featured Interview
- 6 | Software Development
- 8 | Autonomic Computing
- 10 | Robotics

- 12 | Patenting Perspectives
- 13 | Networking and Outreach
- 17 | Partnership Profiles
- 18 | Metrics
- 23 | ICB Awards

Software development is vital to all NASA missions. This has given opportunity for software engineers at Goddard to be innovative in developing leading-edge software code and strategies that address many of NASA's current and future software requirements. Of recent, work has been done in the development of software code that is characteristic of biological processes. One example is the "autonomous nervous system" (which controls basic functions such as breathing). This software technology is designed to help robotic and other software-controlled systems perform more independently and intelligently in very remote and harsh environments.



Nona Cheeks

This issue of Goddard Tech Transfer News looks at several Goddard software technologies. We begin with an interview of Dr. Mike Hinchey, Computer Engineer in Goddard's Computing Environments and Collaborative Technologies Branch (Code 585). In the interview, Dr. Hinchey talks about the work his group is doing in the areas of automated code generation, autonomous computing, and systems that emulate biological processes.

We then take a closer look at how these software technologies can be applied to commercial uses. For instance, one article looks at the robotics market and Goddard technologies designed to allow these devices to be programmed more quickly. We highlight software that enables robots to act more independently to make decisions based on demands of the dynamic environments in which they operate. Another article looks at automated code generation which allows non-programmers to create software code through various forms of input such as natural languages. Dr. Hinchey and his group have also developed a novel technique for automatically generating code more quickly and accurately than existing methods. Additionally, this technology can be used "in reverse" to generate functional descriptions and theoretically, documentation from existing code. We also discuss several autonomic technologies developed by Goddard that mimic biological functions such as heartbeat, pulse, and apoptosis, and how they can be applied to a number of applications involving security, maintenance, and troubleshooting.

In keeping with this issue's software theme, our legal contributors, attorneys Bryan Geurts (Chief Patent Counsel for Goddard's Office of Patent Counsel) and Erika Arner (Partner for the law firm Finnegan, Henderson, Farabow, Garrett & Dunner), offer their perspectives on the patenting of software technologies. They also weigh in on the recent "patent issues" underway in the software industry; and respective benefits and risks of the various ways to protect software, including patent, copyright, and Trade Secret.

If you would like to learn more about Goddard software-related capabilities, technologies, and opportunities to license or partner with NASA Goddard, please contact us at ipp.gsfc.nasa.gov.

Nona Cheeks
Chief, Innovative Partnerships Program Office (Code 504)
NASA's Goddard Space Flight Center

Automated Software Development Technology Transfer Opportunities

Goddard Space Flight Center is currently seeking to transfer its portfolio of patents which covers its automated software development technologies to private industry. This portfolio includes technologies which are applicable to the software development industry, as well as the field of robotics and industrial process controls, to name a few. More broadly these patents hold value in any field where a need exists to design highly complex, automated, and intelligent systems.

Leading the way in the effort to commercialize these technologies is Senior Technology Transfer Manager Darryl Mitchell of the Innovative Partnerships Program Office. Darryl is enthusiastic about the merits and utility of the patent portfolio. “The patent portfolio Goddard has built around Mike Hinchey and his colleagues’ work is truly staggering with regard to the potential scope of its impact,” said Darryl.

He sees the versatility of these patents as one of the primary strengths of the portfolio. “The applications range the spectrum from something as broad, fundamental and profound as changing the paradigm of how software is created, to applications as specific as autonomic management of smoke detector networks in buildings.”

With such a broad range of applicability, this patent portfolio has the potential to make major contributions in terms of both technological and economic development impacts. “I’m looking forward to finding organizations to license these innovations so that they can realize their full potential,” he said.

To discuss how your company can acquire or take advantage of this truly remarkable set of patents, please contact Darryl Mitchell via email at darryl.r.mitchell@nasa.gov.



Darryl Mitchell

Code: 504

Years with NASA: 22

Education: **BS Physics, Rutgers University; MS Applied Physics, Johns Hopkins University; MBA Technology Management, University of Phoenix**



This portfolio of patents can greatly simplify the task of creating and programming adaptive and highly complex systems, such as those found in industrial robotic manufacturing.

Featured Interview

In this issue of GSFC Tech Transfer News, we speak with Dr. Mike Hinchey, Computer Engineer in Goddard's Computing Environments and Collaborative Technologies Branch (Code 585). Dr. Hinchey is former Director of the GSFC Software Engineering Laboratory. He is also the founding editor-in-chief of the NASA journal, Innovations in Systems and Software Engineering. Our discussion with Dr. Hinchey focused on his software work, particularly in the areas of automated code generation, autonomous computing, and systems that emulate biological processes.

Q. How does automated code generation work?

Traditional code generation systems basically attempt to translate informal natural language or graphical input directly into code. This often results in lots of errors and inefficiencies within the coded program. These tools tend to generate large amounts of code that never actually gets used.

Our approach, which is patented, is to incorporate a mathematical model into the process. In our system, we take informal input, such as a natural language description, and use an algorithm to turn it into a formal mathematical model. This model is then converted to code.

It's much easier to derive accurate and efficient code from a formal mathematical model. It also requires significantly less computing power.

Q. What applications do you see for code generation technology?

Quite a few, actually. For example, it allows non-programmers to develop software code. All they need to do is learn how to properly format the natural language or graphical input from which the mathematical model is created. This input could be existing specification material that describes how the code is supposed to work -- in essence, you could create a software product from its functional specification.

This could be an important tool for trained programmers and designers as well. For instance, this could be used as a design tool for a software system. As you design the system, you can run the design through the algorithm, and then assess how the resulting code functions. This allows you to modify and fix the system while it's still in the design stage. Then when the code is working properly, you can either use the generated code, or just code the design yourself. To our knowledge, no other existing code generation tool or approach can do this while guaranteeing equivalence between the code and the specification.

It's important to note that the mathematical model can work "in reverse," so to speak. In other words, you can start with code, run it through the algorithm, and have it generate natural language or graphical output. This would allow a programmer to create a description of pre-existing code. This description could then be used as a functional specification, or even documentation.

There's also numerous applications outside of software product development, such as robotics.

Q. How could this technology be applied to robotics?

It makes programming robots much easier and quicker. Imagine that you have a robotic device that you need to program to do a particular task. In the past, you'd need a software developer to take your instructions and write them as code that the device can understand. Then you'd have to do debugging, to fix whatever issues there may be in the code.

With our system, you can program the device solely through informal input. You can just tell the device what to do, using for example natural language. And this can avoid a lot of debugging time and effort, since you'll be able to see right away whether or not the device is doing what you intended it to do. In large facilities that use lots of interacting robots, such as factory floors and similar industrial settings, this makes design and reconfiguration much easier. All you need is a sufficiently complete description to generate all the required code for each device.

We're also looking at various forms of informal input. For instance, eventually it may be possible to program a robotic device simply by issuing voice commands.

Q. Could this be done in real time, so that for example a physically challenged individual may be able to control a device via voice?

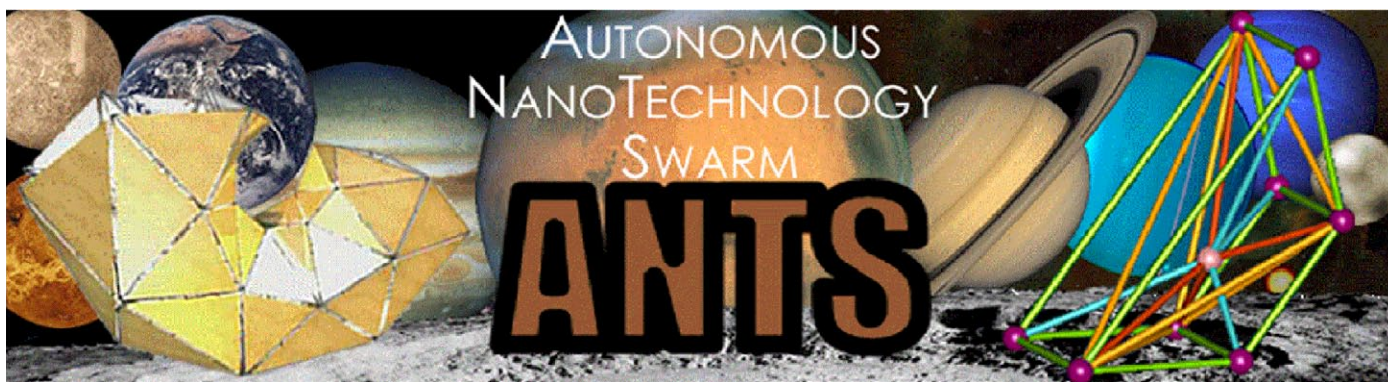
That's actually an interesting idea. We need to look into it!

Q. You've also done quite a bit of work with autonomic computing. Can you describe this briefly?

For autonomic computing, we took a lot of inspiration from biology, especially biological concepts such as heartbeat, pulse, and apoptosis.

Q. Could you explain apoptosis?

"Apoptosis" refers to the fact that cells are pre-programmed to die once they've fulfilled their useful role, unless they get some sort of "reprieve" signal from the body. You can use this concept as a model for managing complex systems with many different components, all of which consume system resources. A component may have a specific function, one that may need to be performed only once, or only in special situations. Rather than have this component remain active and consuming system resources, you could build in a mechanism that mimics apoptosis. If the component does not get a "reprieve" signal within a certain expiration time, it automatically shuts down and no longer is an active part of the system.



This functionality can be an important tool for system management, since it helps ensure that the only active parts of your environment are the components that you actually need -- the ones that aren't useful anymore are automatically taken offline. It's also useful in systems that are highly dynamic and changing, such as battlefield networks. And apoptosis also has value in cyber security. Many environments have unused components hanging around, components that hackers might use to gain unauthorized entry. Apoptosis can ensure that such components can be deactivated when they've served their purpose, thereby closing off these potential vulnerabilities.

Q. How do the concepts of "heartbeat" and "pulse" apply to computing?

A "heartbeat" signal is simply a periodic message that a component issues to the system, ensuring that it is still functioning. Basically, if the system stops receiving a heartbeat signal from a component, it assumes the component is no longer functioning and adjusts accordingly. We use this concept for the ANTS architecture (Autonomic NanoTechnology Swarm, for more information see <http://ants.gsfc.nasa.gov/ArchandAI.html>). When one component of an ANTS swarm stops sending out a heartbeat signal, the swarm assumes it's no longer operational and adapts.

"Pulse" is a little more complicated. It's more or less a heartbeat signal that also contains additional data about the "health" of the component. The pulse can be analyzed, and appropriate corrective measures can be taken if the signal indicates the component is experiencing a problem. The pulse can also indicate whether or not data sent from the component can be considered reliable. For example, if a sensor is reporting some off-the-wall measurements, its pulse can be analyzed to help determine whether or not the data is accurate, or whether it may be affected by some problem the sensor is experiencing. The system can then choose whether to accept the data, or toss it out.

Q. Has private enterprise expressed interest in these technologies?

We have gotten some interest. For example, we've talked to IBM, who was interested in developing a process control tool that they could offer commercially.

Beyond this, any big contractor or developer of large, complex systems should be interested in these technologies.



Michael Hinchey

Code: **585**

Years with NASA: **10**

Education: **BS Computer Systems, University of Limerick; MS Computation/Mathematics, University of Oxford; PhD Computer Science, University of Cambridge; Chartered Engineer; Chartered Mathematician; Chartered Professional Engineer; Chartered IT Professional**

GSFC Code Generation and Software Development

Dr. Mike Hinchey and his colleagues in Goddard Space Flight Center's Computing Environments and Collaborative Technologies Branch (Code 585) have developed a number of technologies designed to greatly reduce the time and effort required to create software programs. These technologies can automatically create formal specifications and working code from informal input. The potential value of such tools to the commercial software community is both significant and obvious.

In this article, we summarize some of the work performed by Dr. Hinchey and his team. We also briefly look at the market for software development tools, and conclude with a list of some of the GSFC inventions that may be of interest to this market.

Code Generation

A number of GSFC inventions involve a novel method for creating formal specifications and code from informal descriptions. Basically, code generation tools allow non-programmers to create code, using various forms of input and interfaces. This input is then translated into functional code. Although such tools have been available for a number of years, the code they produce tends to be inefficient and error-prone. This is largely due to the difficulty in translating informal input, such as natural language, into code. As a result, such tools have yet to revolutionize the software industry; the vast majority of code, even relatively simple programs, still requires the expertise of a professional programmer.

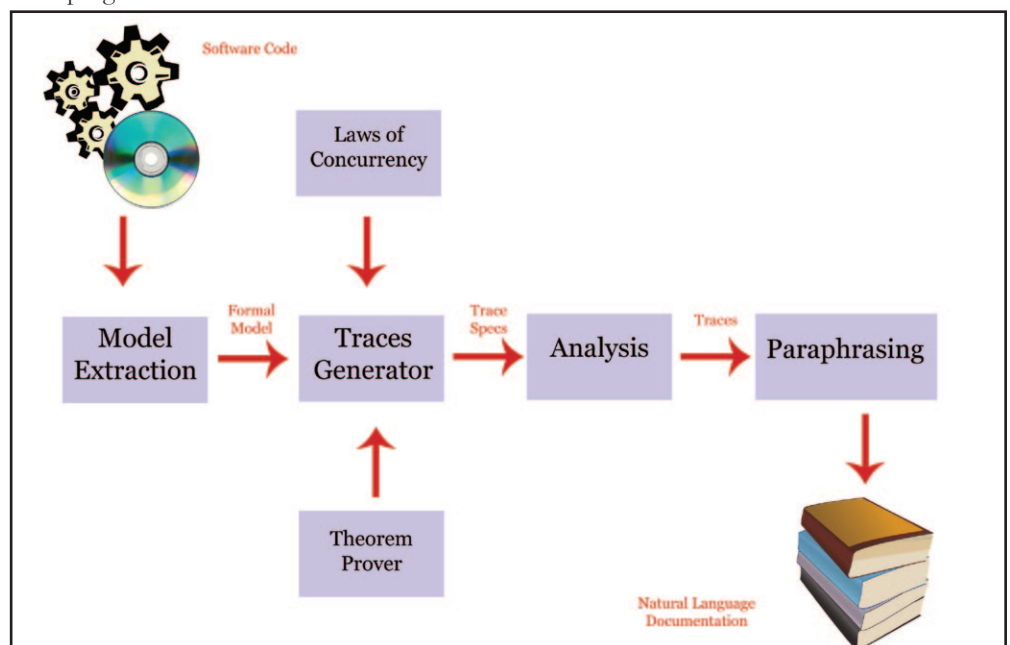
Dr. Hinchey and his group are developing a new approach to code generation. Instead of attempting to translate the informal input directly into code, this new approach incorporates an algorithm that creates a formal mathematical model of the input. This mathematical model can then be converted to code. The process of translating a formal mathematical model into code is much more efficient and accurate than creating code from informal input. It also requires less computing power.

This novel approach to code generation allows non-programmers to create functional, efficient code. The input from which the code is created can be an informal natural language description or flowchart of the code logic. This means that programs could be created from their functional specifications, and that code design and creation can essentially be combined in a single step. This can also be a very useful tool for professional programmers, since they can now simulate their programs during the design stage -- programmers can use the algorithm to create functional code of their early designs, and run the resulting program to determine whether or not it is working as intended. This allows corrections and modifications to be performed during the design stage, before the programmer begins the actual coding of the program. This in turn can significantly reduce debugging, since the program logic will already have been proven beforehand.

"Reverse Engineering" Descriptions from Code

An important feature of GSFC's code generation technology is its ability to "run in reverse" and create a description of the code -- in other words, a programmer can input source code, which will be converted into a formal mathematical model. This model is then translated into a description of the code.

As virtually any software vendor can attest, documentation can be the "critical path" in the overall software development process. Typically, the documentation writer is required to finish the complete documentation by the time the program is debugged, tested, and ready to ship. Frequently the documentation is incomplete or inaccurate, as changes and additions to the program occur throughout the development process. And every time the program is tweaked, the documentation needs to be changed accordingly. As a result, producing and maintaining documentation can be costly, time-consuming, and often painful. GSFC's algorithm may be able to significantly streamline this process. By running the finished code through the algorithm, developers can create natural language descriptions that in some cases could potentially serve as documentation; or at a minimum can be used as the primary source material for the documentation writer. This description also has the advantage of accuracy, since it is produced directly from the code itself. Programmers can also use this method to create descriptions of pre-existing code for which no documentation or functional specifications exist.



This diagram shows how GSFC's code generation technology can be used to reverse engineer software documentation for existing code.

The Software Development Market

It should go without saying that commercial software development is a vast, global, multi-billion dollar industry. Software has become an essential component of the modern world. One segment of the software industry is the market for application design/development tools (often abbreviated as ADT). ADT includes third generation languages, unified development environments, web site design tools, and analysis, modeling, and design tools. The GSFC technologies discussed in this article fall into this last category. The total market for ADT was estimated at almost \$7 billion in 2010, growing at an annual rate of 4.6%.

Key drivers of the ADT market include:

- The need to rapidly develop applications to support changing business needs
- Increased complexity and size of software projects
- Dynamic business environments within enterprises
- Integrating software programs

GSFC's technologies can directly address all four of these drivers. As we've noted, non-programmers can now create code from informal input. This can greatly reduce the amount of development time necessary to produce software programs. And as business needs change, the code can be changed accordingly, simply by changing the informal description. This is also advantageous for developing large, complicated software systems. If the system can be described (either with natural language or graphically) then code can be quickly created for it, irrespective of its complexity. As we've noted, these technologies allow for quick modification of existing code, allowing programs to serve the needs of dynamic environments. And integrating a new program into an existing system in theory could be as straightforward as adding its description to that of the overall system

GSFC Formal Specification and Code Generation Technologies

The following is a brief listing of some of the formal specification and code generation related technologies available for licensing from Goddard Space Flight Center:

Deriving Formal Specifications and Code from Scenarios (GSC-14389-1) and ***A Method and System for Direct Implementation of Formal Specifications Derived Mechanically from Informal Requirements*** (GSC-14941-1) are related to the core technique for mechanically deriving both a formal specification and executable computer code from both English language and programming language representations of scenarios of hardware or software system behavior. The formal specifications provide a means to check the scenarios for potential errors that are difficult or nearly impossible to find with specifications that have no formal underlying mathematical foundation. From the formal specification, this technique automatically generates computer-executable code, which will ensure that no errors are introduced into the code from manual coding.

A Method and System for Formal Analysis, Simulation, and Verification of Knowledge-Based Systems, Rule-Based Systems, and Expert Systems (GSC-14942-1) describes the use of an "inference engine" for converting system rules into a process-based specification that can then be analyzed.

Generation and Verification of Policies for Autonomic Computing Systems (GSC-15079-1) extends the GSFC code generation techniques into the field of autonomic computing. In this invention, policies developed for autonomic computing systems can be verified, to ensure that the policies are both reasonable and complete. The code for implementing these policies can then be automatically generated. The related invention disclosure, ***Modeling, Specifying and Deploying Policies in Autonomous and Autonomic Systems Using an AOSE Methodology*** (GSC-15178-1), applies to using graphical notation based on UML for specifying policies for autonomic systems.

A Method of Deriving Process Based Specifications from Scenarios via Pattern Matching (GSC-15080-1) adds pattern matching to these technologies, which may have significant application in recognizing protein sequences in DNA samples, as well as identifying rogue agents, viruses, or other malicious code in a software system.

An Incremental Approach to System Evolution using R2D2C Requirements Based Programming and Automata Learning (GSC-15331-1) allows for input expressed as a set of scenarios to be converted into a process-based description. This extends the technologies by incrementally supporting system evolution.

Deriving Formal Specifications and Code from Sets of Scenarios, Traces, and/or their Descriptions (GSC-15498-1) allows input in the form of trace descriptions or descriptions of sets of traces, or scenario descriptions or descriptions of sets of scenarios.

Summary

As we've seen, the work of Dr. Hinchey and his team can have a number of practical benefits for multiple facets of software development. Professional developers will appreciate the modeling and simulation capabilities these technologies provide. Non-programmers can take advantage of the fact they can now create working code from informal input. And documentation writers will now have a new tool to help ensure their user documentation and functional specifications are complete and accurate. These capabilities should be of interest to anyone involved in the production of software systems.

Takeaways

GSFC is developing a number of technologies related to code generation. These technologies allow a non-programmer to create code. They also can serve as analysis, modeling, and design tools for commercial programmers. In addition, they could significantly streamline and expedite the process of producing functional specifications and documentation. These tools could offer significant value for the ADT market, a multi-billion dollar industry.

For more information on these and other Goddard software development technologies, please contact the GSFC Innovative Partnerships Program Office (Code 504), at 301-286-5810, or by email at techtransfer@gsfc.nasa.gov, or visit: <http://ipp.gsfc.nasa.gov>

Autonomic Computing

The term “autonomic computing” has become popular in recent years, as a novel and advanced strategy for managing resources in complex systems and networks. Goddard Space Flight Center has developed several autonomic computing concepts, to support missions and applications both on the ground and in space.

In this article, we briefly review basic autonomic computing theory. We then examine several autonomic technologies developed by GSFC, including apoptosis, heartbeat, and pulse. We conclude with a quick listing of several examples of GSFC inventions related to autonomic computing.

Autonomic Computing: A Primer

Autonomic computing is a concept that has been developed by IBM and others over the past 10 years as a way to manage increasingly complex computing systems. The core idea borrows from biology, specifically the human body’s autonomic nervous system. This controls basic bodily functions such as breathing and blood pressure, without any conscious effort or intervention required from the individual.

Autonomous computing models this system. In an autonomic system, basic functions are self-managed in accordance with a pre-defined set of policies and rules established by system administrators and designers. The computer system then uses these policies and rules to manage and maintain itself, without direct action from administrators. Since this minimizes the amount of administrative oversight, autonomic computing can be an important strategy for managing extended and complicated systems, such as those associated with cloud computing.

IBM has defined the following four components of autonomic computing:

- Self-Configuration: Automatic configuration of components
- Self-Healing: Automatic discovery, and correction of faults
- Self-Optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements
- Self-Protection: Proactive identification and protection from arbitrary attacks

GSFC Autonomic Computing Concepts

The Goddard Space Flight Center Computing Environments and Collaborative Technologies Branch (Code 585) has developed a number of technologies that mimic autonomic biological functions. Three examples of these are apoptosis, heartbeat, and pulse.

Apoptosis

The first of these concepts, apoptosis, may at first seem an unlikely inspiration for a computer technology. It involves the pre-programming that instructs cells to die once they have fulfilled their useful life and function. Biologically, this prevents cells that are no longer useful from consuming energy, allowing the body to focus instead on supplying energy to cells that are still performing an important purpose.

GSFC has adapted this concept as a way to help manage large, complicated computer systems. Many of these systems include

components designed to be used only once, or only for a limited period of time, or only in special situations. Allowing these entities to remain part of the system after they have served their purpose can consume resources. To avoid this, components can have an apoptosis feature built in that causes them to shut down and remove themselves from the system if a pre-defined expiration period passes without the component receiving a “reprieve” signal. This allows the system to allocate its resources only to those components that are still playing an active and useful role.

This functionality can be an important tool for system management, since it helps ensure that the only active parts of your environment are the components that you actually need -- the ones that aren’t of value anymore are automatically taken offline. It’s also useful in systems that are highly dynamic and changing, such as battlefield networks. And apoptosis also has value in cyber security. Many environments have unused components hanging around, components that hackers might use to gain unauthorized entry. Apoptosis can ensure that such components can be deactivated when they’ve served their purpose, thereby closing off these potential vulnerabilities.

Heartbeat and Pulse

Two related autonomic concepts developed at GSFC are heartbeat and pulse. Both are designed to help systems communicate with and manage their separate components. Of the two, heartbeat is the simpler concept. It involves a component periodically sending a “heartbeat” signal to the system. This heartbeat signal alerts the system that the component is online and functioning. If the system fails to receive the heartbeat signal, it assumes that the component is no longer operational, and adjusts accordingly -- for example, by assigning another component the tasks originally performed by the lost component.

Heartbeat can be an important tool in systems and networks that consist of multiple objects and entities that are physically distant and disparate. For example, the Autonomic NanoTechnology Swarm, consisting of numerous “nano-components,” uses heartbeat to keep the system informed about which components are still present and functional parts of the overall swarm. Heartbeat signals can also be applied in deployed military networks, where assets are on the move and may be lost and added as needed. And it could also be useful in cloud computing environments, to manage far-flung resources.

Pulse is related to heartbeat, although it adds a layer of complexity. As with the heartbeat, a pulse can let the system know that the component is still operational and communicating. However, the pulse also contains additional data about the overall “health” of the component, such as whether or not the unit is experiencing some problem. The pulse signal can be analyzed, and

if possible the system can attempt to correct (or at least make allowances for) the component's problem. The pulse can also be used to help determine whether or not the unit is functioning properly, or sending reliable data.

GSFC Autonomic Computing Inventions

The following is a brief listing of some of the autonomic computing related technologies available for licensing from Goddard Space Flight Center:

A Biologically-Inspired Method of Improving System Performance and Survivability through Self-Sacrifice (GSC-15550-1) is a biologically-inspired approach to modeling self-optimizing system behavior for NASA swarm-based exploration missions, whereby the individual system entities can sacrifice themselves for the greater good of the entire system. The ANTS concept mission consists of a large number of spacecraft units that can function with a high degree of autonomy, robustly handle gathering-data and computation activities under harsh conditions in space, and configure and reconfigure aspects of an emergent self-optimizing behavior of ANTS, inspired by the self-sacrifice behavior observed in some hive cultures.

Otoacoustic Protection in Biologically-Inspired Systems (GSC-15206-1) mimics the “acoustic reflex” that the mammalian ear uses to protect itself in the presence of loud sounds. In this invention, the “sound” is spurious signals or signals generated by a “rogue agent. The goal is to protect the system by counteracting the noise by generating a signal that would stop autonomic managers from receiving (or at least reacting to) the noise.

An Autonomic Smoke Detector (GSC-15179-1) exploits GSFC's pulse technology (based on a double-heartbeat “lub dub” signal) to autonomically manage smoke detectors in a building. The “lub” part of the signal indicates the health of the unit (such as whether or not it is transmitting data properly, or in a timely fashion). The “dub” part carries instructions

or signals, either triggered by a particular event (such as the detection of smoke) or to perform various operational functions.

Autonomic Quiescence (GSC-15176-1) builds on the apoptosis metaphor that provides a self-destruct signal between autonomous agents. In this approach, a component or craft in a system can have a “half-sleep” (quiescent) state built in that takes the component off-line, but from which the unit can be re-activated by the system if needed. This can be considered a less extreme version of apoptosis self-destruction.

Summary

As computer systems and networks become more complicated and widely distributed, as in cloud computing applications, autonomic computing concepts will likely play an increasingly larger role in their management. The GSFC autonomic computing technologies discussed in this article, as well as others currently in development, may represent opportunities for the development of commercial tools that could bring significant value to the computer system management and administration community.

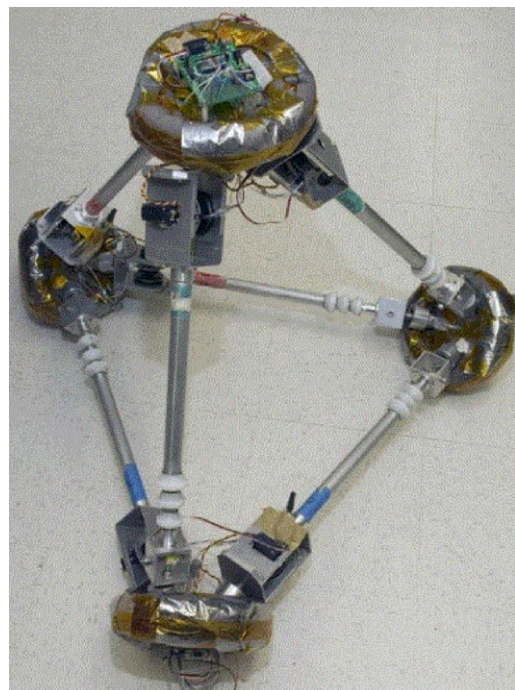
Takeaways

Autonomic computing is a concept that borrows from biological processes. Autonomic features allow complex computer systems to acquire some level of self-management, based on policies and rules set by administrators. GSFC autonomic technologies include apoptosis, heartbeat, and pulse. These are designed to help manage disparate components in complicated distributed networks.

For more information on these and other Goddard autonomic computing technologies, please contact the GSFC Innovative Partnerships Program Office (Code 504), at 301-286-5810, or by email at techtransfer@gsfc.nasa.gov, or visit: <http://ipp.gsfc.nasa.gov>



The Tetrahedral Walker (right) relies on GSFC software to maneuver autonomously over harsh terrain. It was tested and proven at McMurdo Station in Antarctica (above).



Robotics and Software Development

Robots play a critical, in fact virtually ubiquitous role in modern space exploration. Numerous robots are currently serving in space, performing in all facets of exploration including Earth science, deep-space exploration, and Solar Systems missions.

To control these robots and ensure they perform at peak efficiency, Goddard Space Flight Center's engineers and scientists have developed numerous software technologies. Some of these are designed to more quickly and effectively create programs to control robotic devices. Others are intended to allow robots to act with more autonomy, making decisions based on the demands of the dynamic environments in which they operate. This ability to act autonomously is especially important in planetary exploration, where great distances preclude instantaneous control from Earth. And as with other GSFC inventions, many of these software technologies also have significant commercial potential for terrestrial applications, especially within the rapidly expanding world of robotic devices.

In this article, we'll take a quick look at the robotics market. We'll then examine several GSFC software technologies that could represent attractive licensing opportunities within this market.

World Robotics Market Overview

A recent report from the Web site CompaniesAndMarkets.com estimates the global market for robotic devices at approximately \$21 billion. This same report projects that this market will grow to \$30 billion by 2016, with an annual CAGR of 6.7% during this period. Fastest growth will be in Asia, where the market is expected to grow at a 7.2% CAGR, reaching almost \$11 billion by 2016. The North American market is also substantial, projected to reach \$5 billion by 2016 (although growing at a more modest 2.7% CAGR).

The robotics industry consists primarily of six major segments. These include space robots, industrial robots, military robots, security robots, domestic service robots, and professional service robots.

- Space robots often take the form of manipulation arms under human control. A common application is the unloading of the docking bay of space shuttles to launch satellites or to construct a space station. Another genre of space robots includes robotic probes used for planetary exploration. These probes, which have been deployed by NASA since the early 1960's, often require a great deal of autonomy and decision-making capability, which frequently takes the form of artificial intelligence.
- Industrial robots are typically used in manufacturing processes, such as automobile factories. Common tasks for these devices include welding, cutting, lifting, sorting, and bending. Many of these technologies are also being applied for smaller-scale processes such as food processing, where robots perform trimming, cutting, and processing of various meats. In addition, industrial robots are used commercially for exploring and monitoring environments dangerous to human observers. Robots are also being developed for agriculture. These include automated harvesters for cutting and gathering crops, as well as robotic dairies that allow farmers to remotely feed and milk their cows.

- Military robots include airborne surveillance drones. In the near future, robots will also likely be deployed for applications such as automated aircraft and vehicles, for applications such as carrying supplies or clearing minefields.
- Security robots monitor restricted environments. Their role is to detect unauthorized intrusion and take appropriate action, for instance sounding an alarm or automatically securing valuable or vulnerable assets.
- Domestic service robots include so-called "intelligent home" automated systems that monitor home security, environmental conditions and energy usage. These devices can also open doors and windows; control lighting, heating, and air conditioning; control appliances, and perform similar tasks.
- Professional service robots are devices designed for environments such as hospitals. For example, a robotic suit is under development that will enable nurses to more easily lift patients. Robots can also perform surveillance, using advanced sensing and imaging equipment to operate in hazardous environments such as areas damaged by earthquakes or other natural disasters. Robotics can also be used in education.

These segments comprise a widely diverse market in which technologies designed to make it easier to control these robots -- or allow them to control themselves -- could be in high demand.

Robotic Control and Code Generation

One key GSFC technology that makes programming robots much easier and quicker is automated code generation, developed by Mike Hinchey and the Computing Environments and Collaborative Technologies Branch (Code 585). As Mike explains, the "traditional" method for programming a robotic device to do a particular task involves developing code in a programming language understood by the robot. This can be a complicated and time-consuming process, especially when debugging and redesign are factored in.

With automated code generation technology, a non-programmer can issue instructions to a robot through informal input such as natural language or graphics. The code generation algorithm then converts this informal input into a formal mathematical language, which in turn is used to generate actual code. In this way, users can create code to control robotic devices, without any special knowledge of the programming language involved, quickly and relatively easily. The redesign and bugging processes will also be greatly accelerated, since users can assess the performance of the robot without having to wait for the coding to be completed by a programmer. This also makes system reconfiguration, such as adding new components or instructing a device to perform a different action, much easier. This allows robotic processes to be set up more quickly, for example to manufacture a limited run of specially modified units.

Robotic Autonomy and Artificial Intelligence

Another important area for GSFC software development is autonomy. As explained in the article “Autonomic Computing” in this issue of *GSFC Tech Transfer News* autonomic computing entails the development of systems that mimic biological functions such as heartbeat and breathing. Autonomic functions allow systems, including those that control robotic devices, to perform tasks and automatically respond to environmental variables without direct intervention from humans.

In addition, GSFC has developed several artificial intelligence (AI) technologies applicable to robotic devices. A number of these AI features have been developed with space missions in mind, where ground-based control is inconvenient and/or impossible due to the distances involved. AI could be equally valuable here on Earth, especially for environments (such as the deep sea, high radiation areas, structurally compromised enclosed spaces, and so on) where human control would either be too slow or technologically infeasible.

The “Autonomic Computing” article discusses several GSFC inventions related to autonomic computing. Below are several other GSFC inventions related to AI; these could have potential in terrestrial robotic applications.

Sensor Complete Requirements Algorithm For Autonomous Mobility (SCRAAM) (GSC-15527-1) is an algorithm that determines the sensor requirements for autonomous mobility. SCRAAM examines the time evolution of a given mobility-related behavior in terms of its sensitivity, both to classes of sensors as well as the defining physical characteristics of those sensors. SCRAAM can be used to determine the sensor requirements of any synthetic neural system, either mobile or immobile.

Stability Algorithm for Neural Entities (SANE) (GSC-15357-1) is an algorithm for determining the stability of synthetic neural systems. The purpose of SANE is to identify “psychological instabilities” in these systems. This involves examining how the AI reacts to a broad range of conditions, and flagging areas in which the system performs in a way that significantly deviates from its “normal” behavior. Such large-magnitude changes, invoked too quickly, could lead to instability and the eventual collapse of the entire system. By correcting these “over-optimizations,” an AI system could be designed to be more reliable and “human like.”

Evolvable Neural Software System (ENSS) (GSC-14657-1) is composed of sets of Neural Basis Functions (NBF's) which can be autonomously created and removed according to the changing needs and requirements of the ENSS. The resulting structure of the ENSS is both hierarchical and self-similar in that a given set of NBF's may have a ruler NBF, which in turn communicates with other sets of NBF's. A number of commercial applications exist for this technology, including spacecraft, avionics systems, robotics, security systems, and information systems.

Summary

These and other GSFC technologies can be adapted to a wide range of potential uses, including applications far removed from the space science missions for which they were originally developed. For example, the SANE algorithm forms the basis of the Asymptotic Diet Algorithm with Psychological and Temporal Stability (ADAPTS), a “psychologically and temporally stable” diet in which behavioral changes are relatively small and implemented slowly, thereby falling within the SANE criteria for stability. This exemplifies how, with a little creativity, GSFC technologies can provide the foundation for a broad spectrum of commercially viable products.

Takeaways

GSFC has a great deal of experience developing software for controlling robotic devices. These include automated code generation that allows for quicker programming of robots by non-programmers. GSFC software technologies also include autonomous systems and artificial intelligence; both allow robots to respond on their own and make decisions without direct human intervention. These technologies have potential commercial value within the terrestrial robotics market, in a variety of applications including industrial, military, security, and others.

For more information on these and other Goddard robotic technologies, please contact the GSFC Innovative Partnerships Program Office (Code 504), at 301-286-5810, or by email at techtransfer@gsfc.nasa.gov, or visit: <http://ipp.gsfc.nasa.gov>

Patenting Perspectives

In this edition of Patent Perspectives, we look at the ins and outs of patenting software technologies. Among the topics discussed are the ongoing “patent wars” between some of the major players in the industry. We also examine the issue of copyright vs. patent as it pertains to software.

Offering their perspectives on these subjects are attorneys Bryan Geurts (Chief Patent Counsel for GSFC’s Office of Patent Counsel) and Erika Arner (Partner for the law firm Finnegan, Henderson, Farabow, Garrett & Dunner).



Bryan Geurts

Q. How “patentable” is software?

Bryan: Patents are typically not the best mechanism to protect software. Simply transforming a way of doing something into code does not a patent make. On the other hand, some software does rise to the standard of patentable subject matter. For instance, software code -- such as that being developed by Mike Hinchey -- is based on algorithms or methods. Algorithms can often be patented, even if all code cannot.



Erika Arner

Commercial code is often protected by copyright. However, this does not always help the U.S. government, because government work completed solely by civil servants cannot be protected by copyright within the U.S. (although outside the country, such copyrights can be enforced for U.S. government content).

Erika: I wouldn’t say that software is not patentable. Although it is true that the code itself is not patentable, usually it’s protected by copyright. Unfortunately, copyright in this case offers only very limited protection.

Most software patenting involves protecting the functionality of the software, which provides stronger protection. In some cases, source code may be included in the patent document, but that’s relatively rare. If you patent the functionality and features of the software, that prevents programmers from reverse-engineering your program.

Q. How does patenting software compare to copyrighting it, or protecting it via Trade Secret?

Erika: It’s more difficult to obtain a patent compared to copyright. The patent process requires you to prove that you’ve come up with something new. The only requirement for copyright protection is that it’s an original work, but the underlying ideas do not have to be new for copyright protection. Any code can therefore be copyrighted.

But as we’ve said, copyright protection is rather limited. It’s relatively easy for a programmer to simply reverse-engineer a software program and write slightly different code that provides all the same functionality of the original. So this limits copyright protection; essentially all it does is prevent someone from simply copying and pasting the original code.

Bryan: You can copyright your code, and still maintain your software as Trade Secret. When you register the copyright on the code, you’re only required to include the first ten and last ten pages of the source code. Everything else in the middle can be kept secret. That’s why programmers are often advised not to put anything important in the first ten or last ten pages of anything they plan to copyright.

Erika : Trade Secret is an important way to protect software, especially because it can be difficult to detect when one program infringes on the protected intellectual property of another. To prove infringement, for example, you have to show that someone else’s program uses the same algorithm you’ve patented. This can be difficult to prove conclusively. So many developers decide to protect their programs under Trade Secret laws instead of patents.

Bryan : Most of the big commercial software companies out there protect their products under Trade Secret. For example, Microsoft and Apple use this to protect their operating systems.

Q. Where does Open Source fit it?

Bryan: When you release code as Open Source, you’re essentially giving up control of it. This means that your code can evolve in any number of directions. In some respects, that’s a good thing -- it’s a good way to get code out there “free and clear.”

Open Source is becoming an increasingly important part of the commercial software world. Not so much among companies such as Microsoft and Apple, although they too are doing some Open Source stuff. But companies such as Google and Mozilla are governed by open standards.

Q. You mentioned the work of Mike Hinchey earlier. Why is GSFC pursuing multiple patents for this code, rather than going the Open Source route?

Bryan: Basically, what Mike Hinchey is doing represents a new way to develop software -- the code writes itself, so to speak. The user defines the requirements, standards and boundaries, and the technology creates the actual code based on these. This is really a new approach for writing software, based on a specific algorithm rather than the actual code itself. Therefore it doesn’t really lend itself to Open Source.

Q. There’s been a lot of recent news about software companies buying up patents and taking each other to court over them. Is there some type of competitive advantage to doing this?

Erika: There’s a couple of things going on here. It certainly can be a competitive advantage when you buy up a huge block of patents; it can make you an instant competitor within a market in which you might not have had any previous presence. Others buy up patents so they can license them and obtain additional revenues.

Another reason for doing this is self-protection. Some consortiums are buying up patents simply to take them off the market. This reduces the risk that they’ll fall into the hands of a litigious entity. Thus some patent acquisitions are more a defensive move.

Bryan: A lot of litigation among the big players in the software industry involves the Android platform, which currently has

around 40% of the wireless market. The big companies are competing fiercely for this market, so there's a lot at stake. A lot of software-related litigation would probably go away if Android were not so dominant.

Erika : It's a little akin to the old VHS versus Betamax battle. Smartphones represent a huge market. If adopters end up moving towards one standard, some of the others will likely become obsolete. So companies are aggressively trying to protect their investments.

Q. Where is software patentability headed in the near term?

Erika : The U.S. Patent Office is assuming a wait-and-see posture in regards to software patenting. There are a number of software-related patent cases currently in the Federal Circuit Court of

Appeals. Within the next 6 to 12 months, there should be a number of decisions issued involving software patentability. The situation is in a bit of flux right now.

Bryan: I think Erika's being generous when she says there's some "flux" at the moment. The word I'd use is "turmoil." Anyone who touches software is very worried about the current situation, waiting anxiously to get some guidance from someone.

Erika: I agree. The question is, who will provide that guidance, the Patent Office, Congress, or the Courts? It's like in the Ferris Bueller movie: Who's going to speak up? Anyone? Anyone?

Readers, what patent issues would you like to have Brian and Erika discuss in future issues? Please send suggestions to lucy.a.stefanelli@nasa.gov.

Business Networking and Outreach

STS-135 Shuttle Launch and OPTIMUS PRIME Event

(July 6 – 10, 2011, Orlando FL)

NASA Goddard Space Flight Center's (GSFC) Innovative Partnerships Program Office (IPPO) hosted an exhibit at the Kennedy Space Center's launch celebration of STS-135 on July 8, 2011. These exhibit efforts included displays on technology transfer, NASA spinoff technologies, the NASA OPTIMUS PRIME Spinoff Award Contest featuring Peter Cullen, the voice behind Hasbro's TRANSFORMERS brand main character, OPTIMUS PRIME. The event included a showing of the TRANSFORMERS 3 movie and also featured an interview with Sci-Fi Authors featuring a science fiction writing challenge.



Above: Innovative Partnerships Program Office (IPPO) staff member, Adil Anis, hosts the NASA Spinoff Prize Wheel game, giving attendees the opportunity to win a prize for correctly answering NASA Spinoff Technology Trivia.



Left: The voice of OPTIMUS PRIME, actor Peter Cullen, takes time to autograph a drawing for one of his many fans.

Business Networking and Outreach

What's Your Favorite Space?

(August 17th, 2011, New York, NY)

The Innovative Partnerships Program Office (IPPO) played a central role in the success of the “What’s Your Favorite Space?” event at the Eventi hotel plaza in New York on August 17th, 2011. As part of a large event celebrating NASA, the IPPO provided a walking gallery of Hubble images, actual examples of NASA spinoffs behind glass, demos of the multiplayer online game Moonbase Alpha, and videos running on a 35-foot screen above the plaza.



Innovative Partnerships Program Office (IPPO) staff member Adil Anis, demonstrates the multiplayer online game Moonbase Alpha, to attendees of the “What’s Your Favorite Space?” event in New York City.



Innovative Partnerships Program Office (IPPO) staff member Melissa Jackson explains how NASA technologies are used in our homes and in our everyday lives to attendees.

Emerging Space and Commercial Suborbital Vehicles Workshop

(September 7 – 8, 2011, Washington, DC)

NASA Goddard Space Flight Center’s (GSFC) Innovative Partnerships Program Office (IPPO) and Earth Science Division hosted a Commercial Suborbital Vehicles Workshop at GSFC on September 7, 2011. The purpose of the workshop was to provide information for Earth and Space scientists about vehicles’ capabilities and examined and discussed science topics that might be conducted from these platforms. The key opportunities enabled by the unique capabilities of the commercial suborbital vehicles are that they would provide frequent and low-cost access to the mesosphere,

lower thermosphere (MLT) region of the atmosphere (50-140km altitude). Atmospheric observations from commercial suborbital vehicles would provide a new window on the MLT. This workshop allowed for an open discussion of platform capabilities versus measurement needs, allowing for any potential design requests to be easily accommodated.



Innovative Partnerships Program Office (IPPO) staff member, Trina Cox, hosts the Next Steps 2011 registration table.



2011 Next Steps Conference attendees meet and discuss the SBIR/STTR program during one-on-one sessions with NASA representatives.

Next Steps in Managing Innovation Workshop

(September 29th, 2011,
Pittsburgh, PA)

NASA's Goddard Space Flight Center's Innovative Partnerships Program Office (IPPO) hosted the Annual Mid-Atlantic Next Steps in Managing Innovation Workshop on September 29, 2011 at the DoubleTree Hotel in downtown Pittsburgh, Pennsylvania. This event gave SBIR/STTR companies the opportunity to become acquainted with the technology needs of NASA Goddard as well as network with NASA technologists, SBIR Phase IIIs, prime contractors, and other partners that have unique needs and methodologies for advancing technology. This event was a great opportunity to learn more about the various partnering mechanisms and resources available to small businesses to support technology development. Other highlights included information on SBIR data property rights, use of NASA Intellectual Property in proposals and working with NASA to promote successes.



Innovative Partnerships Program Office staff member Jennifer Geiger, presented at the 2011 Next Steps Conference. Her presentation entitled, "The SBIR/STTR Program Perspective" emphasized the importance of having a champion for your technology as well as the importance of Phase III funding and commercialization.

Business Networking and Outreach

networking and outreach



2011 SPIE Optics + Photonics Conference

(August 21 -25, 2011, San Francisco, CA)

Innovative Partnerships Program Office (IPPO) staff members Enidia Santiago-Arce and Brent Newhall attended the 2011 SPIE Optics + Photonics Conference held at the San Diego Convention Center from August 21 – 25, 2011. The Conference is the largest inter-disciplinary technical conference in North America. NASA Goddard's IPPO hosted a booth detailing the science and technology behind the James Webb Space Telescope. IPPO personnel were on hand to talk about their ongoing "Can You See It Now?" Campaign. This effort is focused on transitioning a deep portfolio of wavefront sensing and adaptive optics technologies, procedures, and lab equipment to private industry.

Innovative Partnerships Program Office (IPPO) Technology Manager Enidia Santiago-Arce, discusses wavefront sensing technologies with an SPIE conference attendee.

Space Entrepreneurship Forum

(September 21 – 24, 2011, Washington, DC)

The Goddard Innovative Partnerships Program Office supported the Space Entrepreneurship Forum on September 21, 2011 at Rayburn House Building Room in Washington D.C. The forum was coordinated by JAKA Consulting Group and Juxtopia LLC which is a NASA partner. The private space industry is estimated to become close to a one trillion dollar industry by 2020. This forum raises awareness of space entrepreneurial opportunities and areas of growth within the African American business community. The Goddard IPPO Chief Nona Cheeks was a member of the industry panel at the forum.



Partnership Agreements July- September 2011

The IPPO is pleased to announce the recent signing of these partnership agreements.

partnership profiles

Partner	Type	NASA Goals/Benefits
Tor Books New York, NY	Non-Reimbursable Space Act Agreement	<ul style="list-style-type: none"> GSFC has partnered with Tor-Forge Books to develop and publish a series of “NASA Inspired Works of Fiction,” based on concepts pertinent to current and future agency missions and operations. These books are intended to introduce, inform and inspire readers about NASA. NASA will pair scientists and engineers with Tor-Forge writers to help raise awareness and enhance public interest in science, technology, engineering and mathematics (STEM). NASA’s goal is to attract and retain student interest in STEM studies, strengthening the agency and the nation’s future workforce.
NSF Center for High-Performance Reconfigurable Computing (CHREC) Gainesville, FL	Non-Reimbursable Space Act Agreement	<ul style="list-style-type: none"> Testing of the SpaceCube v1.0 board. Bridge the gap between NSF CHREC’s laboratory experiments and flight ready experiments that can be uplinked to the on-orbit SpaceCube board demonstration platform.
DoD STP	Reimbursable Space Act Agreement	<ul style="list-style-type: none"> Integration of the Space Test Program - Houston 4 (STP-H4) payload. DoD STP to provide integration and flight of the GSFC International Space Station SpaceCube Experiment 2.0 (ISE 2.0) on STP-H4. GSFC to provide two (2) flight qualified units of the SpaceCube Communication Interface Box (SpaceCube CIB) for STP-H4.
Xilinx San Jose, CA	Non-Reimbursable Space Act Agreement	<ul style="list-style-type: none"> GSFC to conduct an independent heavy ion evaluation of the radiation performance of the Xilinx XQR5VFX130-CF1752 integrated circuit. Device is the state-of the-art radiation hardened by design (RHBD) field programmable gate array (FPGA) built on a highly scaled technology process. Potential game changing device for space system design will simplify design requirements (i.e., reduced additional radiation mitigation) while providing high level system performance and reconfigurability.
DoD STP	Ammendment-Reimbursable Space Act Agreement	<ul style="list-style-type: none"> Continue the work done for the integration of the Space Test Program - Houston 4 (STP-H4) payload.

New Technology Reports: 215

V-Assembly Dual Head Efficiency Resonator (VADER) Laser Transmitter by Barry Coyle, Paul Stysley, and Demetrios Poullos (Code 554)

Multiwalled Carbon Nanotube Absorber Coupled with YBCO Superconducting Thermometer for Infrared Detection by John Hagopian (Code 551) and Patrick Roman (Code 553)

Impedance Matched to Vacuum, Invisible-Edge Diffraction Suppressed Mirror by John Hagopian (Code 551), Edward Wollack (Code 665), and Patrick Roman (Code 553)

Dynamically Reconfigurable Systolic Array Accelerator by Robert Barnes and Aravind Dasu (Code 551)

Silicon Carbide Vacuum Ultraviolet Photodiode Arrays by Alexander Bolotnikov and Larry Rowland (Code 553)

A Novel High-Performance Micropump by Mahmooda Sultana and Sachidananda Babu (Code 553)

Inductive Power Transfer for Spaceflight Systems by Michael Wright (Code 500)

Reconfigurable, Wideband Radar Transceiver and Antenna for P-band Stretch Processing by Feng Xu, Arvind Bhat, Yik-Kiong Hue, and Joon Chun (Code 555)

Low-Field Vector Helium Magnetometer by Robert Slocum and Andy Brown (Polatomic, Inc)

Application of Uncertainty Visualization Methods to Meteorological Trajectories by Ryan Boller, David Laidlaw, Jadrian Miles (Code 587), and Scott Braun (Code 613)

SmallSat Constraint and Deployment System by John Hudeck and Luis Santos (Code 548)

Low Cost Variable Conductance Heat Pipe for Balloon Payloads by Calin Tarau (Code 820)

CMOS Compatible SOI MESFETs for Radiation Hardened DC-to-DC Converters by William Lepkowski, Seth Wilk, and Trevor Thornton (Code 563)

Cloud Liquid Water Content Sensor with Droplet Sizing Capability by John Bognar (Code 614.6)

Motion-Corrected Three-Dimensional Sonic Anemometer for Tethersondes and Other Moving Platforms by John Bognar (Code 614.6)

Flight Test Device for Very Small Unmanned Aircraft Systems by John Bognar (Code 614.6)

Process and Design Improvements for Picometer Resolution MEMS Deformable Mirrors by Michael Helmbrecht (Code 667)

Cloud Absorption Radiometer (CAR) by Michael King (Code 610)

Development of Optical Parametric Amplifier for Lidar Measurements of Trace Gases on Earth and Mars by Kenji Numata (Code 663), Haris Riris (Code 694), Stewart Wu, Michael Krainak (Code 554), Stephan Kawa (Code 613.3), and James Abshire (Code 690)

BACAR (BRDF (Bidirectional Reflectance Distribution Function), Albedo, Aerosol Radiometer) by Charles Gatebe (Code 613.2), Bert Paquale (Code 551), Jason Budinoff (Code 544), and James Smith (Code 550)

Phase Change Material for Temperature Control of Loop Heat Pipe Compensation Chamber by Michael Choi (Code 545)

The Primordial Inflation Explorer (PIXIE) by Michael Seiffert (Code 326.2), Alan Kogut, David Chuss, Jessie Dotson, Dale Fixsen, Mark Halpern, Gary Hinshaw, Stephan Meyer, Samuel Moseley, David Spergel, Edward Wollack (Code 665)

Anti-Biofouling for Ocean Instruments and Floats Using Microchlorine Generation by George Seaver (Code 614.6)

Space Weather Android App; A Standalone Android Application Which Displays Space Weather Information to Users by David Berrios, Justin Boblitt, Richard Mullinix, Richard Maddox (Code 587), and Michael Hesse (Code 696)

Hardware for Accelerating N-Modular Redundant Systems for High Reliability Computing by Carl Dobbs and Keith Bindloss (Code 500)

Pass Plan Formatter (PPF) for Earth Sciences Ground Support System (TERRA and TRMM) by John Kerich and James Busch (Code 444)

Iris-Shutter Cube by Robert Thate (Code 547)

Miniaturized Laser Heterodyne Radiometer for Carbon Dioxide (CO₂), Methane (CH₄), and Carbon Monoxide (CO) Measurements in the Atmospheric Column by Emily Steel (Code 694) and Matthew McLinden (Code 555)

CODAT (Composition and Dynamics Analysis Tools) by Tilak Hewagama (Code 693)

Development of the Hilbert-Huang Transform Real-Time Data Processing System with 2-D Capabilities by Semion Kizhner (Code 564)

Utilization of Waste Heat to Warm Optical Telescopes or Lasers in Space by Using Loop Heat Pipes by Michael Choi (Code 545)

Radio Frequency Plasma Discharge Lamps for Use as Stable Calibration Light Sources by Brendan McAndrew (Code 551), John Cooper (Code 614), Angelo Arcucci, Greg McKee, and Christopher Durell (Labsphere.com)

Mid to Far Infrared Optimized Carbon Nanotubes for Stray Light Suppression by Stephanie Getty (Code 541), John Hagopian, Manuel Quijada, and John Rigueur (Code 551)

Mid to Far Infrared Optimized Carbon Nanotubes for High Efficiency Instrument Calibrators by John Hagopian, Manuel Quijada (Code 551), and Stephanie Getty (Code 541)

Sequential Probability Ratio Test for Spacecraft Collision Avoidance Maneuver Decisions by James Carpenter (Code 595) and Francis Markley (Code 591)

Graphene Transparent Conductive Electrodes for Next Generation High Density Focal Plane Assemblies by Mary Li and Mahmooda Sultana (Code 553)

Parallelization of Snowflake Growth Simulation by Christopher Pearson (Code 423) and Thomas Clune (Code 610.3)

Linear Covariance Analysis of Epoch State Estimators by Francis Markley (Code 591) and James Carpenter (Code 595)

Lateral Transfer, Recirculating Fabry-Perot Etalon Spectrometer by Michael Krainak, Mark Stephen, and Molly Fahey (Code 554)

Fabrication of Low Noise Equivalent Power Microwave Kinetic Inductance Detectors for Sub-millimeter Astronomy by Ari Brown, Kevin Denis, Thomas Stevenson, George Manos (Code 553), Kongpop U-yen (Code 555), and Samuel Moseley (Code 685)

2D Microstrip Antenna Feed Array to Multimode Parallel Plate Waveguide by Wei-chung Huang (Code 567), Kongpop U-yen (Code 555), Samuel Moseley, and Edward Wollack (Code 665)

Elastic Deployable Composite Tubular Roll-Out Boom by Brian Spence and Steve White (Code 674)

Gradient Doped Ceramic Nd:Yag Rod Laser Gain Modules by Barry Coyle, Paul Stysley, and Demetrios Poullos (Code 554)

Microwave Kinetic Inductance Detector with Selective Polarization Coupling by Edward Wollack (Code 665), Kongpop U-yen (Code 555), Thomas Stevenson, Ari Brown, Wen-Ting Hsieh (Code 553), and Samuel Moseley (Code 685)

COLLADA Computing for Geophysical Applications by Scott Shipley (Code 610.2)

Nitinol-Actuated Normally Open Permanent Isolation Valve by Daniel Ramspacher and Caitlin Bacha (Code 597)

Domain Engineered Magnesium Oxide Doped Lithium Niobate for LIDAR-based Remote Sensing by Philip Battle, Carsten Langrock, and Martin Fejer (Code 564)

Broadband Planar Antenna using Multi-branch Slotline by Kongpop U-yen (Code 555), Samuel Moseley (Code 685), Edward Wollack (Code 665), and Kevin Denis (Code 553)

A COBE/DIRBE Point Source Catalog: Investigating the Infrared Variability of Asymptotic Giant Brant Stars by Beverly Smith (East Tennessee State Univ.)

SpaceWire Packet Error Handling by Michael Pagen (Code 560)

Glass Solder Approach for Robust, Low Loss Fiber to Waveguide Coupling by Shirley McNeil, John Lower, Robert Wiley, Philip Battle, Brett Clark, and Todd Hawthorne (Code 562)

Manufacturing and Test Methods to Produce Low-Scatter Spectral Bandpass Filters by Peter Fuqua, Thomas Mooney, and John Figoski (Code 551)

TechCube CubeSat by Jacqueline LeMoigne-Stewart (Code 580) and Thomas Flatley (Code 587)

Software Interface with Hardware by Manuel Buenfil, Thomas Flatley (Code 587), and Jacqueline LeMoigne-Stewart (Code 580)

Goddard Mission Services Evolution Center (GMSEC) Application Programming Interface (API) MATLAB Wrapper 1.0 by LaMont Ruley, Robert Wiegand (Code 583), Jacqueline LeMoigne-Stewart, and Joseph Yan (Code 580)

Micro-scale Electrohydrodynamic (EHD) Thermal and Fluid Management Device by Timothy Miller (Code 553) and Jeffrey Didion (Code 545)

A High Repetition Rate Seeded Optical Fiber Amplifier (SOFIA) for the LIST mission and Next Generation Satellite Laser Ranging by Demetrios Poullos, Richard Kay, Donald Coyle (Code 554), and Gordon Blalock (Code 694)

Multi-Gigabit Rate Radiation Hard Bus by Vladimir Katzman (Code 561)

Fabrication Process for Infrared Blocking Filters for X-Ray Astrophysics by James Chervenak, Ari Brown (Code 553), and Edward Wollack (Code 665)

Ultra-Low Noise Quad Photoreceiver for Space-Based Laser Interferometric Gravity Wave Detection by Abhay Joshi, Shubhashish Datta, and Jim Rue (Code 663)

CANS (CAR Autonomous Navigation System) by Charles Gatebe, Bill McCune, Dustan Hellwig (Code 613.2), and Duncan Kahle (Code 553)

Functionalized Nano-Film Microchannel Plate: A Single High Aspect Ratio Device for High Resolution, Low Noise Astronomical Imaging by Neal Sullivan and Philippe de Rouffignac (Code 660)

Silicon Microchannel Plate Large Area UV Detector by Paul Shnitser, Roman Ostroumov, Alexander, and Usikov Soukhoveev (Code 660)

Low-Noise, UV-to-SWIR Broadband Photodiodes for Large-Format Focal Plane Arrays by Abhay Joshi (Code 663)

A MEMS and Carbon Nanotube Detector Mount for Yttrium Barium Copper Oxide (YBCO) High Temperature Superconductor Chips by Patrick Roman (Code 553)

Matrix: GMSEC Component Supporting Plugins by Robert Wiegand, LaMont Ruley (Code 583), Jacqueline LeMoigne-Stewart, and Danford Smith (Code 580)

Improved Rock Core Sample Break-off, Retention and Ejection System by Thomas Myrick (Code 544)

All-Spherical Surface, All-Reflective Cassegrain Version of the MicroMak Star Sensor Maksutov-Bouwers Optical Design by Albert Simpson and John Collins (Code 596)

VLBI Antenna Calibration by GPS by Daniel MacMillan and Thomas Clark (Code 698)

High Accuracy Cryogenic Heat Flow Measurement Device for Cryogenic Testing (Q-meter) by Stuart Glazer (Code 545)

FPGA-Based X-Ray Detection and Measurement for an X-Ray Polarimeter by Kyle Gregory, Joanne Hill, Wayne Baumgartner (Code 564), and Kevin Black (Code 662)

Bismuth Passivation Technique for High Resolution X-Ray Detectors by James Chervenak and Larry Hess (Code 553)

Process for Patterning Indium for Bump Bonding by Kevin Denis (Code 553)

Small Submersible Robust Microflow Cytometer for Quantitative Detection of Phytoplankton by Mark Dugan and Ali Said (Code 610)

Planar Waveguide Laser With Graded Reflectivity Output Facet by Demetrios Poullos, Barry Coyle, and Paul Stysley (Code 554)

Integrated Ground Supported Systems (IGSS) Equation Processor by Esther Woodward, LaMont Ruley, Jane Steck (Code 583), Jacqueline LeMoigne-Stewart (Code 580), Thomas Pfarr, and Jim Langston (Code 444)

Java-based Kameleon Converter: Conversion Software to Convert Space Weather Simulation Data to Standardized Data Formats by David Berrios, Marlo Maddox, Nitesh Danti, Thomas Flatley (Code 587), Jacqueline LeMoigne-Stewart (Code 580), and Lutz Rastaetter (Code 674)

A 10kWatt 36GHz Solid-State Power Amplifier using GaN-on-Diamond by Felix Ejeckam, Dubravko Babic, Daniel Francis, and Firooz Faili (Code 555)

Independent Testing Capability Testing and Simulation Framework by Steven Seeger, Justin McCarty, Brandon Bailey, Shawn Carroll, Jeff Joltes, Dan Nawrocki, David Soto, and Justin Morris (Code 180)

Twin Head Efficient Oscillator (THEO) Development for the ACE Multi-Beam Lidar and 3D-Winds by Donald Coyle, Paul Stysley, Richard Kay, Demetrios Poullos (Code 554), Robert Frederickson, Gordon Blalock, and Kenneth Cory (Code 694)

Low-Loss All-Dielectric Reflectors for Far-Infrared Applications by Ari Brown, Kevin Denis (Code 553), Manuel Quijada (Code 551), S. Harvey Moseley (Code 660), Edward Wollack, and Alexander Kuttyrev (Code 665)

Thermoformed Contamination Barrier by Edward Freymiller and Kevin Malone (Code 551)

Integrated Science Instrument Module (ISIM) Reconfigurable Packet Tool by Esther Woodward, LaMont Ruley, Jane Steck (Code 583), Jacqueline LeMoigne-Stewart (Code 580), Jim Langston, and Thomas Pfarr (Code 444)

Application of Advanced Filtering Techniques to Current Flight Projects and Future Missions by Joel Parker (Code 595)

scLinux - Embedded Linux Distribution for SpaceCube by Gary Crum, Thomas Flatley (Code 587), and Jacqueline LeMoigne-Stewart (Code 580)

SMART Flight Software by Karin Blank (Code 586), Jacqueline Le Moigne-Stewart (Code 580), Thomas Flatley (Code 566), Peyush Jain (Code 588), Jennifer Sichler, and Gary Crum (Code 587)

Simple Subset Wizard by Christopher Lynnes, Edward Seiler, and Mahabaleshwara Hegde (Code 610.2)

A Distributive, Non-Destructive Real-Time Approach to Snowpack Monitoring by Jeff Frolik and Christian Skalka (Univ. of Vermont)

Flexible, Lightweight Vacuum Shell for Load Responsive MLI for High Thermal Performance In-Air and On-Orbit by Scott Dye (Code 552)

Environmental Qualification of Single Crystal Silicon Mirror for Space Flight Use by John Hagopian, John Chambers, Scott Rohrbach (Code 551), Vincent Bly (Code 553), Jason Budinoff, and Armando Morell (Code 544)

RF Reference Switch for Space Flight Radiometer Calibration by Joseph Knuble (Code 555)

Interoperable Remote Component (IRC) Architecture Version 7 by Troy Ames, Tom Flatley, Jeffrey Hosler (Code 587), Carl Hostetter (Code 588), and Jacqueline LeMoigne-Stewart (Code 580)

Space Weather Explorer 2: A Java-based Visualization Program for Space Weather Simulation Data by David Berrios, Marlo Maddox, Thomas Flatley (Code 587), Jacqueline LeMoigne-Stewart (Code 580), and Lutz Rastaetter (Code 674)

An Innovative Active Heater Control Concept to Meet IXO Type Mirror Module Thermal-Structural Distortion Requirement by Michael Choi (Code 545)

Module::Build::Database by Brian Duggan, Phil Durbin, and Curt Tilmes, (Code 614.5)

Reflectance Reduction Techniques for Ultra-Black Infrared Coatings by Edward Wollack (Code 665)

A Non-Intrusive Method of Resolve the Thermal-Dome-Effect of Pyranometers by Si-Chee Tsay and Qiang Ji (Code 613.2)

Tech Transfer Metrics July - September 2011

Kameleon-Plus: Data Access, Interpolation, and Fieldline Tracing Library for Space Weather Simulation Data by David Berrios, Thomas Flatly, Marlo Maddox (Code 587), Jacqueline LeMoigne-Stewart (Code 580), and Lutz Rastaetter (Code 674)	Software to Simulate Data Flow from the Interface Data Processing Segment (IDPS) to the Science Data Segment (SDS) for the NPOESS Preparatory Project by Chiu Yeung (Code 585), Jacqueline LeMoigne-Stewart (Code 580), and LaMont Ruley (Code 583)	Application of the Computational Framework to Synthesize and Simulate Radiometer Architectures by Paul Racette (Code 555), Tom Clune (Code 610.3), and Damon Bradley (Code 564)	Graphene-GaN Schottky Photodiodes by Mahmooda Sultana, Mary Li, James Chervenak (Code 553), and Shahid Aslam (Code 693)	Radar Range Sidelobe Reduction using Adaptive Pulse Compression Technique by Lihua Li, Michael Coon, and Matthew McLinden (Code 555)
High efficiency Multiwalled Carbon Nanotube Radiator by John Hagopian, Manuel Quijada (Code 551), and Stephanie Getty (Code 541)	JWST MSS Project Thermal Chamber GSE Automated Control by Dion Smallwood (Code 586)	Airborne Lidar Measurements of Atmospheric Pressure Made Using the Oxygen A-Band by Haris Riris, Michael Rodriguez, Graham Allan, William Hasselbrack, Mark Stephen, and James Abshire (Code 694)	GMSEC API WebLogic Middleware Wrapper by Matthew Handy (Code 583)	The Use of a Frequency-Offset Pulse Train for Surface Range Sidelobe Compensation for Nadir-Pointing Pulse Compression Radar by Matthew McLinden, Lihua Li (Code 555), and James Carswell (Code 556)
Lunar Night Time Power Source by Peter Chen (Code 671)	Oscilloscope Waveform Capture Program by Alyson Topper, Hak Kim, Jonathan Pellish (Code 561)	X-Ray Diffractive Optics by Brian Dennis (Code 671), Gerald Skinner (Code 661), and Mary Li (Code 553)	Radiation Design of Ion Mass Spectrometer for Europa by Edward Sittler (Code 673)	Fabrication of Valve Seat-Block Components of an Ultra Low Leak Piezoelectrically Actuated Microvalve by Manuel Balvin, Yun Zheng, Mary Li (Code 553), and Eui-Hyeok Yang (Code 384.1)
Integrated Space Weather Analysis System (iSWA) by Marlo Maddox, Thomas Flatly, David Berrios (Code 587), Jacqueline LeMoigne-Stewart (Code 580), Richard Mullinix (Code 541), Peyush Jain (Code 588), Michael Hesse (Code 696), and Lutz Rastaetter (Code 674)	Time-Multiplexed Pipelined FFT Implementation by Rafael Garcia (Code 564)	Computational Framework for Analyzing Data Using Calibrated Noise by Paul Racette (Code 555)	Visible-Infrared Hyperspectral Image Projector by Matthew Bolcar (Code 551)	GMSEC API WebLogic Wrapper by Matthew Handy (Code 583)
Secure Password Storage (Controlled) by Glenn Bock (Code 444)	Dynamic Threshold Voltage Control for Readout Integrated Circuits by Todd Grimes (Code 564)	Instantaneous Doppler Correction Dual Laser FMCW Laser Radar by Mina Rezk and Anthony Slotwinski (Code 551)	Atomic Layer Deposition to Enable the Production, Optimization and Protection of Valuable Spaceflight Hardware by Vivek Dwivedi (Code 545)	Computer Modeling for Generation of Synthetic Radio Reflection-Transmission Tomography (RRTT) Data for Exploring Subsurface Geology of Asteroids and Comets by Manohar Deshpande (Code 555)
Metamaterial Absorbers for Terahertz Bolometer Applications by Timothy Miller and I-Kuan Lin (Code 553)	Method for Reducing Spectroscopic Interferences in Trace Gas Measurements by Mark Paige (Code 612)	Multi-Channel RF GPS Receiver Integrated Circuit by Jack Kennedy and Michael Shaw (Code 596)	Aerogel-Based Multi-Layer Insulation with Micrometeoroid Protection by Redouane Begag and Shannon White (Code 545)	High Collection Efficiency Fluorescence Detection Cell by Thomas Hanisco, Maria Cazorla, and Andrew Swanson (Code 613)
Precision Navigation Strategies for Primitive Solar System Body Sample Return Missions by Kenneth Getzandanner (Code 595) and John Adams (Code 596)	Towards Efficient Scientific Data Management Using Cloud Storage by Qiming He (Code 417)	Digital Design Training Module by Renee Reynolds (Code 561)	GMSEC API WebSphere MQ Wrapper by Matthew Handy (Code 583)	Software Defined Radio with Parallelized Software Architecture by Greg Heckler (Code 567)
Software to Compare NPP HDF5 Data Files by Chiu Yeung (Code 585), Jacqueline LeMoigne-Stewart (Code 580), and LaMont Ruley (Code 583)	CEDI Instrument Design Development for the GEO-CAPE Mission by James Smith (Code 550), Catherine Marx, Oscar Martinez (Code 551), Jason Budinoff (Code 544), and Antonio Mannino (Code 614.2)	External Cavity Laser by Jordan Camp (Code 663)	Wide Band, Dual Polarized Ultra-Low Noise Focal Plane Array Feed For Active/Passive Microwave Remote Sensing by Manohar Deshpande (Code 555)	Fabrication and Application of Nano Geometries as Optical Elements by Patrick Roman (Code 553)
Microelectronic Repair Techniques for Wafer-Level Integration by Larry Hess and David Franz (Code 553)	Development of a Low-SWaP, RAD-Tolerant, Thermally Stable High Speed Fiber Optics Network for Harsh Environment Applications by Matt Leftwich, Tony Hull (Code 561), and Fred Orlando (Code 560)	Power Provision Based on Self-Sacrificing Spacecraft by Michael Hinchey and Emil Vashev (Code 585)	Absolute Wavelength Control of Lasers for Active Sensing in Space by Sarah Bickman, Michael Anderson, Scott Davis, and Benjamin Luey (Code 554)	A System and Method for Generating Power In a Dam by Rene Carlos (Code 543)
Cleaning Telescope Mirrors with Electron Beams by Fred Minetto (Code 695)	Use of Temporal Algorithms and Calibrated Noise to Characterize Receiver Stability by Paul Racette (Code 555)	GeNTRe, Version 1 by Luther Lighty (Code 585)	Thermal Control with FPGA-based PI Controllers and Buck Regulators by Matthew Owens (Code 560), Michael Edick (Muniz Engineering Inc.), Edward James, and Jose Florez (Code 564)	
		3D Finite Element Model of Magnetic Mirror and Nano Antenna by Shahram Shiri (Code 551)	Conventional Batch Microprocessing for Sub Micron Nano Optical Components by Patrick Roman (Code 553)	
		GMSEC API WebSphere MQ Middleware Wrapper by Matthew Handy (Code 583)		
		Autonomic Self-Esteem by Roy Sterritt and Michael Hinchey (Code 581)		

New Solar Panel and System Design Features that Reduce Heating and Optimize Corridors to Allow for Lower Risk Planetary Aerobraking

by Jill Prince (Code D205), Michael Amato (Code 101), Charles Baker (Code 592), Derek Liechty (Code D305), David Steinfeld (Code 545), Jeff Stewart (Code 543), and John Dec (Code D206)

MMS Spacecraft Flight Software

by Mark Walters, Daniel Berry, Tawanda Jacobs, Ronald Miller, Michael Yang, Joel Chiraló, Peter Kutt, and Alexander Schoening (Code 582)

Atmospheric Plasma Deposited Films for Space Stable TCMS with High Leakage Current Capabilities

by Mukund (Mike) Deshpande and Edward Pierson (Code 546)

Integrating SpaceCube with ST-5

by Larry Kepko (Code 674)

A Method, System, and Apparatus for Providing Water to a Heat Engine via a Dammed Water Source

by Rene Carlos (Code 543)

Efficient Reasoning with Ambient Trees for Space Exploration

by Michael Hinchey (Code 581) and Emil Vassev (Code 585)

Autonomic and Apoptotic Cloud, Autonomic and Apoptotic Grid, Autonomic and Apoptotic Highly Distributed System

by Roy Sterritt and Michael Hinchey (Code 581)

Autonomic and Apoptotic Security Systems

by Roy Sterritt and Michael Hinchey (Code 581)

A Method, System, and Apparatus for Powering a Compressor Via a Dam

by Rene Carlos (Code 543)

Apoptotic Robotics

by Roy Sterritt and Michael Hinchey (Code 581)

Soft X-Ray Filter with Improved IR Rejection and Improved Durability

by Bruce Lairson (Code 662)

High Performing PFPE Nanofluid Lubricants

by Bryan Bergeron, Amy Stevens, and Kyle Roberts (Code 541)

Silicon Strip Detectors for Extremely Low Energy Particles

by George Manos, Larry Hess, James Chervenak, Thomas Stevenson, and Anh La (Code 553)

Efficient Space Exploration Through Laziness

by Michael Hinchey and Emil Vassev (Code 585)

Apoptotic Data and Digital Objects

by Roy Sterritt and Michael Hinchey (Code 581)

Technology Independent RHBD Library through Gate Array Approach

by Xiaoyin Yao, Paul Eaton, David Mavis (Code 561)

Autonomic Autopoiesis

by Roy Sterritt and Michael Hinchey (Code 581)

RUSHMAPS: Real-time Uploadable Spherical Harmonic Moment Analysis for Particle Spectrometers

by Adolfo Figueroa-Vinas (Code 673)

Scheduling Operations for Massive Heterogenous Clusters

by John Humphrey and Kyle Spagnoli (Code 762)

Airborne Lidar Measurements of Atmospheric Methane from a High Altitude Aircraft

by Haris Riris (Code 694), Kenji Numata, Steve Li, Stewart Wu (Code 554), and Martha Dawsey (Code 551)

Formaldehyde Profiler using Laser Induced Fluorescence Technique

by Guangkun Li (Code 614)

Virtual Climate Data Server (vCDS) Repetitive Provisioning

by John Schnase, Savannah Strong, and Glenn Tamkin (Code 606)

Software Engineering Tools for Scientific Models

by Marc Abrams, Pallabi Saboo, and Mike Sonsini (Code 610.3)

Boron Nitride, Nano Tube and Nano Mesh for Tailored Conductivity TCMS Space Applications

by Mukund Deshpande (Code 546), Dr. Narayan Hosmane, and A. Chakrabarti (Northern Illinois Univ.)

The Mobile Chamber

by Gregory Scharfstein and Russell Cox (Code 551)

Autonomic Adhesion

by Roy Sterritt and Michael Hinchey (Code 581)

Virtual Climate Data Server (vCDS) Administrative Extensions

by John Schnase and Glenn Tamkin (Code 606)

Virtual Climate Data Server (vCDS) Concept, Design, Architecture, and Operation

by John Schnase and Glenn Tamkin (Code 606)

Remote Memory Access Protocol Target Node Intellectual Property

by Omar Haddad (Code 560)

Virtual Climate Data Server (vCDS) NetCDF Module

by John Schnase, Glenn Tamkin, Savannah Strong (Code 606), and W. David Ripley (Code 583)

Scalable ParaView for Extreme Scale Visualization

by Utkarsh Ayachit, Homa Karimabadi, and George Zagaris (Code 586)

Advanced Pulse Compression System and Testbed

by James Carswell (Code 556)

Digital Beamforming Interferometry

by Rafael Rincon (Code 614.9)

Broadband Ultra Low Power Signal Processing for High Performance Instrumentation - TRL Level Increase

by Damon Bradley (Code 564)

Basic Prototype Tuning System to Modify a Pre-Existing Antenna Design

by Timothy Rink (Code 555)

Planar Superconducting Millimeter-Wave Channelizing Filter

by Negar Ehsan (Code 555), Kongpop U-yen (Code 550), Ari Brown (Code 553), Edward Wollack (Code 665), and Samuel Moseley (Code 660)

Advanced Communications Receiver Processing Methods

by Gregory Heckler, Haleh Safavi (Code 567), and Luke Thomas (DO3/Flight)

A Method for Integrating a Microwave Radiometer into Radar Hardware for Simultaneous Data Collection between the Instruments

by Matthew McLinden and Jeffrey Piepmeyer (Code 555)

Double Parity Single Error Correction (DPSEC) Code

by Wing Lee and Wai Fong (Code 567)

A MEMS Glow Discharge Plasma Electron and Ion Source

by Patrick Roman (Code 553)

V-Assembly Dual-head Efficient Resonator (VADER) for Space-Based Lidar and Altimetry

by Donald Coyle and Paul Stysley (Code 554)

A High Speed Data Recorder for Space, Geodesy and Other High Speed Recording Applications

by Mikael Taveniku (Code 698)

FY11 IRAD #35 Standard Bus for Picosatellites / Wallops CubeSat Processor

by Christopher Lewis (Code 569)

A Compact Radar Transceiver with Included Calibration

by Matthew McLinden and Rafael Rincon (Code 555)

Variable Telemetry Rate System for Satellite and Flight Communications Systems

by Wayne Powell, Rob Tye, Serhat Altunc, and Joseph O'Brien (Code 569)

A Spacecraft Housekeeping System-on-A-Chip (SOC) in a Radiation Hardened Structured ASIC

by George Suarez, Jeffrey DuMonthier, George Suarez, Wesley Powell, and Robyn King (Code 564)

ELID Grinding of Large Aspheres

by Flemming Tinker (Code 551)

Low-Weight, Durable, and Low-Cost Metal Rubber Sensor System for Ultra Long Duration Scientific Balloons

by Andrea Hill (Code 820)

Software to Simulate Data Flow from the Comprehensive Large Array-Data Stewardship System (CLASS) to the Science Data Segment (SDS) for the NPOESS Preparatory Project

by Chiu Yeung (Code 585), Jacqueline LeMoigne-Stewart (Code 580), and LaMont Ruley (Code 583)

CMOS Total Ionizing Dose Sensor and Compensator Circuit

by Gerard Quilligan (Code 564) and Shahid Aslam (Code 693)

Modeling and Design Framework for Silicon Carbide Avalanche Photodiodes

by Akin Akturk, Siddharth Potbhare, Neil Goldsman, and Marc Dandin (Code 553)

Micro Pulse Detonation Rocket Engine for Nano-Satellite Propulsion

by Kenneth Yu (Univ. of Maryland - College Park)

ExPRESS Logistics Carrier (ELC) Flight Controller Unit (FCU) by Kevin Ballou, Kenneth Li (Code 561), George Jackson, Wesley Powell (Code 560), Austin Lanham (Code 596), Julian Ramirez (Code 563), Louise Bashar (Code 581), Patrick DeMent (Code 540), and Ed James (Code 564)

New Wind Measurement Capability Using High-Altitude Imaging Wind and Rain Profiling Airborne Radar (HIWRAP) by Gerald Heymsfield (Code 612), Lihua Li, Daniel Shaubert, Matthew McLinden, Justin Creticos, Manuel Vega (Code 555), James Carswell (Code 556), and Martin Perrine (Code 567)

Radiation Hardened By Design CMOS Bias Circuit by Gerard Quilligan (Code 564) and Shahid Aslam (Code 693)

A Computational Framework to Synthesize and Simulate Radiometer Architectures by Paul Racette (Code 555), Damon Bradley (Code 564), and Tom Clune (Code 610)

Deployable Radial Imaging for V, E, Ne (DRIVEN) by Thomas Moore, David Knudsen, Glyn Collinson (Code 670), Adrienne Beamer (Code 505), David Durachka, Englin Wong (Code 564), Joseph Generie, Alphonso Stewart (Code 543), and Paul Rozmarynowski (Code 690)

Frequency Agile Mid-IR Source for Planetary Exploration by Hansjurg Jost, James Scherer, and Joshua Paul (Code 693)

Broadband Planar Impedance Transformer by Negar Ehsan (Code 555)

Duplicate of An Ultra Low Power Cryo-Refrigerator for Space by Mark Zagarola and Roger Hill (Code 552)

Circularly Polarized Antenna with WiFi Applications for Dextre Pointing Package (DPP) by Victor Marrero-Fontanez, Ken Hersey, Aaron Dabrowski, and David Green (Code 567)

Enhanced MgF2 Overcoated Al Mirrors for FUV Space Astronomy by Manuel Quijada, John Mentzell (Code 551), and Stephen Rice (Code 510)

Cryogenic Liquid Sample Acquisition System Designs for Remote Space Applications by Paul Mahaffy, Melissa Trainer, Douglas Hawk, Christopher Johnson (Code 699), Don Wegel (Code 552), and Michael Amato (Code 101)

Nanocomposite Polymers for Index Matched Immersion Lenses for Infrared Detectors by Kevin Denis, Samuel Moseley, Edward Wollack (Code 665), Larry Hess, and Manuel Balvin (Code 553)

Silicon Black Nanomatrix Light Absorbing Surface Treatment Process for Microfabricated Optical Components by Patrick Roman (Code 553)

Patents Issued: 6

A Method Of Deriving Process Based Specifications From Scenarios Via Pattern Matching by Michael Hinchey (Code 585), Christopher Rouff (Code 500), and James Rash (Code 588)

Hardware And Technique For Dead End Welding Of All Types Of Tubing by Michael Wilks (Code 597)

An Active, Solid-state, 3-Dimensional Range Imaging System by James Blair (Code 694), Vibart Scott (Code 690), and Luis Ramos-Izquierdo (Code 551)

Pseudo-Noise Code Modulation using Return to Zero pulses for Ranging, Altimetry and Communications by Xiaoli Sun (Code 924) and James Abshire (Code 690)

Modeling, Specifying And Deploying Policies In Autonomous And Autonomic Systems Using An AOSE Methodology by Michael Hinchey, Joaquin Pena, and Roy Sterritt (Code 585)

Directed Flux Motor Utilizing Concentric Magnets and Interwoven Flux Channels by Neil Parikh, Andrew Punnoose, Andrew Wilson, and Katherine Strausser

Patent Applications: 16

Nanostructure Secondary Mirror Apodization Mask for Transmitter Signal Suppression in a Duplex Telescope by John Hagopian, Shahram Shiri (Code 551), Jeffrey Livas (Code 663), Stephanie Getty, (Code 541), June Tveekrem, and James Butler

Carbon Nanotubes on Titanium Substrates for Stray Light Suppression by John Hagopian, Manuel Quijada (Code 551), and Stephanie Getty (Code 541)

Molecular Adsorber Coating by Sharon Straka, Wanda Peters, Jack Triolo, Mark Hasegawa, Randy Hedgeland, Kevin Novo-Gradac, Alfred Wong, John Petro, Cory Miller, and Nithin Abraham (Code 546)

Lateral Kevlar Suspension Device (LKSD) by Donald Wegel

Imaging System Aperture Masks for Image Plane Exit Pupil Characterization by Brent Bos (Code 551)

Multicolor Detectors for Ultrasensitive Long-Wave Imaging Cameras by Ari Brown, James Chervenak (Code 553), Dominic Benford, and Edward Wollack (Code 665)

Prototype Genomics Based Keyed-Hash Message Authentication Code Protocol by Harry Shaw and Sayed Husseing (Code 567)

Programmable High-Rate Multi-Mission Receiver for Space Communication by Thomas Drago

Ultra-low Power (< 100mW), 64-Channel Pulse Data Collection System by Duncan Kahle, Federico Herrero (Code 553), and Hollis Jones (Code 555)

Graphite Composite Panel Polishing Fixture by Carl Strojny, John Hagopian (Code 551), and Jason Budinoff (Code 544)

An All-metal, Solderless Circularly Polarized Microwave Antenna Element with Very Low Off-Axis Cross-Polarization by Cornelis Du Toit (Code 555) and David Green (Code 567)

An Improved Method of Fabricating Single Crystal Silicon Light Weight Mirrors by Vincent Bly (Code 553)

Otoacoustic Protection In Biologically-Inspired Systems by Roy Sterritt and Michael Hinchey (Code 585)

HEXPANDO Expanding Head for Fastener Retention Hexagonal Wrench by John Bishop (Code 540)

Extreme Environment Low Temperature Transistor Models by La Vida Cooper (Code 564)

An Instrument Suite for the Vertical Characterization of the Ionosphere-Thermosphere System from 100 km to 700km Altitude by Theodore Finne, Hollis Jones, Andrew Nicholas, and Federico Herrero (Code 553)

Provisional Patents Issued: 3

SpaceCube Mini by Michael Lin (Code 561), David Petrick, Alessandro Geist, and Thomas Flatley (Code 587)

Optimizing Automated Interference Mitigation Scheduling Method and Algorithm by James Rash (Code 585)

Laser Radar Through the Window (LRTW) Coordinate Correction Software by David Kubalak, Joseph Hayden, Theodore Hadjimichael, Bente Eegholm, Raymond Ohl, Randal Telfer (Code 551)

Patent Application Awards: 5

Automatic Extraction of Planetary Image Features by Jacqueline LeMoigne-Stewart (Code 580)

Determining Phase Retrieval Sampling from the Modulation Transfer Function by Bruce Dean (Code 551)

System and Method for Multi-Scale Image Reconstruction Using Wavelets by Bruce Dean (Code 551)

System and Method for Command and Data Handling Space Flight Electronics by Willaim Yuknis, Noosha Haghani, Scott Pursley, and Omar Haddad (Code 561)

Molecular Adsorber Coating by Kevin Novo-Gradac and Cory Miller (Code 546)

Tech Brief Awards: 5

A DNA-Inspired Encryption Methodology for Secure Mobile Ad-Hoc Networks (MANET) by Harry Shaw (Code 567)

Microgravity Enhanced Stem Cell Selection by Jagan Valluri and Pier Paolo Claudio (Code 500)

Imaging System Aperture Masks for Image Plane Exit Pupil by Brent Bos (Code 551)

Launch Method for Kites in Low Wind or No Wind Conditions ("EZLaunch") by Ted Miles (Code 569) and Geoffrey Bland (Code 614.6)

Miniature Laser Magnetometer (MLM) by Andy Brown and Robert Slocum (Code 504)

Board Action Awards: 6

Iterative-Transform Phase-Retrieval Utilizing Adaptive Diversity by Bruce Dean (Code 551)

Miniaturized Double Latching Solenoid Valve by James Smith (Code 544)

Sensor Web 2.0 by Troy Ames, Gregg Rabdeau, Patrice Cappelaere, Robert Sohlberg, Stuart Frye, Steve Chien, Ashley Davies, Daniel Tran, and Joshua Doubleday, Mark Johnston, Steven Schaffer, Donald Sullivan, Frederick Policelli, David McLaren, Daniel Mandl, Vuong Ly, Linda Derezinski, Matthew Handy (Code 581)

Generic Reusable Aerospace Software Platform (GRASP) by Rodney Davis and Susannah Davis (Code 589)

Telemetry and Science Data Software System 1.0 by Liang Hong (Code 614.2) and Lakesha Bates (Code 567)

Weka to Web Coverage Processing Service Translator by Justine Rice (Code 581)

Software Release Awards: 14

Goddard Mission Services Evolution Center (GMSEC) VCR by Thomas Grubb (Code 583)

Sensor Web 2.0 by Patrice Cappelaere, Robert Sohlberg, Stuart Frye, Steve Chien, Ashley Davies, Daniel Tran, and Joshua Doubleday (Code 581)

The Invasive Species Forecasting System - Programs/SWLR by John Schnase (Code 606)

The Invasive Species Forecasting System - Core Services (iCore) by John Schnase (Code 606)

The Invasive Species Forecasting System - Command Interpreter (iShell) by John Schnase (Code 606)

Refinement of the HSEG Algorithm for Improved Computational Processing Efficiency by James Tilton (Code 606)

Campaign Manager (GeoBPMS) by Patrice Cappelaere (Code 581)

Identity Management Service for SensorWebs by Patrice Cappelaere (Code 581)

Open Geospatial Consortium (OGC) Compatible Publish/Subscribe Service - Basic [OPSB] by Patrice Cappelaere (Code 581)

EO-1 Sensor Planning Service [EO-1 SPS] by Patrice Cappelaere (Code 581)

EO-1 Sensor Observation Service [EO-1 SPS] by Patrice Cappelaere (Code 581)

Web Coverage Processing Service (WCPS) by Patrice Cappelaere (Code 581)

Weka to Web Coverage Processing Service Translator by Justine Rice and Daniel Mandl (Code 581)

Flood Dashboard by Patrice Cappelaere (Code 581)

Call for NASA Tech Briefs Magazine "Tech Needs" Article Ideas

NASA Tech Briefs (<http://www.techbriefs.com>) is a monthly magazine with a readership of approximately 450,000 that features the latest NASA-developed technologies. The magazine also features a "NASA Tech Needs" article that addresses a technology need for which NASA is seeking a solution.

Goddard Space Flight Center is continuously seeking ideas for Tech Needs articles to submit for publication in *Tech Briefs*. This is a great opportunity for Goddard researchers to reach a broader audience to find viable solutions from industry and academia.

If you are a Goddard civil servant or contractor and have an idea for a Tech Needs article, please e-mail your idea to Brent.P.Newhall@nasa.gov. For the submission, please summarize in 400 words or less:

1. A statement of the problem that requires a technology solution
2. Solutions previously explored
3. Specific applications that would benefit from this "new" technology

Feel free to include any graphics or pictures that would help explain the technology need.



Goddard Tech Transfer News <http://ipp.gsfc.nasa.gov>

Chief: Nona Cheeks
(301) 286-5810
Nona.K.Cheeks@nasa.gov

Goddard Tech Transfer News is the quarterly magazine of the Innovative Partnerships Program Office (Code 504) at NASA's Goddard Space Flight Center. Also available online at: <http://ipp.gsfc.nasa.gov>. Send suggestions to Robert.S.Leonardi@nasa.gov.